NPS52-90-019

NAVAL POSTGRADUATE SCHOOL
Monterey, California

DEMOM - A MEDIA OBJECT MODEL INCORPORATING NATURAL LANGUAGE DESCRIPTIONS FOR RETRIEVAL SUPPORT

Bernhard Holtkamp
Vincent Y. Lum
Neil C. Rowe

February 1990

Approved for public release; distribution is unlimited.

Prepared for:

Naval Postgraduate School
Monterey, California 93943

NAVAL POSTGRADUATE SCHOOL
Monterey, California

This report was prepared for the Naval Ocean Systems Center and funded by the Naval Postgraduate School.

This report was prepared by:

Bernhard Holtkamp

BERNHARD HOLTKAMP
Adjunct Research Professor
Department of Computer Science

Reviewed by:

Robert B. McGhee

ROBERT B. MCGHEE
Chairman
Department of Computer Science

Released by:

G E Schacher

G. SCHACHER
Dean of Faculty
& Graduate Studies

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release;
distribution is unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
NPS52-90-019

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Computer Science Dept.
Naval Postgraduate School

6b. OFFICE SYMBOL *(if applicable)*
52

7a. NAME OF MONITORING ORGANIZATION
Naval Ocean Systems Center

6c. ADDRESS *(City, State, and ZIP Code)*
Monterey, CA 93943

7b. ADDRESS *(City, State, and ZIP Code)*
San Diego, CA 92152

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
Naval Postgraduate School

8b. OFFICE SYMBOL *(if applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
O&MN, Direct Funding

8c. ADDRESS *(City, State, and ZIP Code)*
Monterey, CA 93943

10. S^URCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
DEMOM - A MEDIA OBJECT MODEL INCORPORATING NATURAL LANGUAGE DESCRIPTIONS FOR RETRIEVAL SUPPORT

12. PERSONAL AUTHOR(S)
Holtkamp, Bernhard, Lum, Vincent Y. and Rowe, Neil C.

13a. TYPE OF REPORT
Progress

13b. TIME COVERED
FROM Ap 89 TO Dec 89

14. DATE OF REPORT *(Year, Month, Day)*
1990 February

15. PAGE COUNT
27

16. SUPPLEMENTARY NOTATION
[illegible]

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Cooperative systems, database integration, transaction concepts, decentralized systems, S-transactions

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

The DEMOM media object data model aims at providing a uniform framework for managing different types of media data, i.e. images, text, sound or graphics. According to DEMOM media objects are defined as a class hierarchy of objects, i.e. images, text, sound, and graphics being subtypes of the general type media object. Representation specific objects are regarded as subordinate types of the corresponding subtype, e.g. a SUN raster image in pixrect format is an instance of the subtype pixrect which is in turn a subtype of image.

Using images as an example we discuss the media object hierarchy, the corresponding access operations and implementation issues. Content oriented search of media data on the basis of predicate calculus is considered as an essential part of DEMOM and hence discussed as well.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Bernhard Holtkamp

22b. TELEPHONE *(Include Area Code)*
(408) 646-2251

22c. OFFICE SYMBOL
52Hk

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

DEMOM - A MEDIA OBJECT MODEL INCORPORATING NATURAL LANGUAGE DESCRIPTIONS FOR RETRIEVAL SUPPORT

Bernhard Holtkamp
Vincent Y. Lum
Neil C. Rowe

Naval Postgraduate School
Department of Computer Science
Code 52
Monterey, CA 93943
U.S.A.

Abstract

The DEMOM media object data model aims at providing a uniform framework for managing different types of media data, i.e. images, text, sound or graphics. According to DEMOM media objects are defined as a class hierarchy of objects, i.e. images, text, sound, and graphics being subtypes of the general type media object. Representation specific objects are regarded as subordinate types of the corresponding subtype, e.g. a SUN raster image in pixrect format is an instance of the subtype pixrect which is in turn a subtype of image.

Using images as an example we discuss the media object hierarchy, the corresponding access operations and implementation issues. Content oriented search of media data on the basis of predicate calculus is considered as an essential part of DEMOM and hence discussed as well.

1.0 INTRODUCTION

Considerable efforts in data engineering have been put into developing DBMS for standard commercial applications, such as accounting, banking and others that basically handle alphanumeric data. These conventional DBMS, however, do not provide the functionality that is required in forthcoming nonstandard applications like office automation or computer integrated manufacturing, for instance [Sh88].

In addition to the alphanumeric data the nonstandard applications require the management of non-alphanumeric data like images, sound, graphics and text, generally referred to as media data. The combination of alphanumeric and media data is not only necessary for these applications but is a **must** for enabling the management of media data themselves [Yo88]. The raw media data like a pixel matrix of an image, for instance, are completely useless without some alphanumeric data, generally called registration data, that contain information about the encoding technique used, the colormap and the like.

As a consequence, the underlying data model of a media DBMS has to provide means for combining alphanumeric and media data on two levels. Firstly, a direct combination of these two kinds of data is necessary in order to make up a usable instance of media data, as the alphanumeric information, defining how to interpret the raw data, is needed for interpretation. Secondly, alphanumeric data is frequently required to be combined with media data from the application point of view. An example of such a combination is combining picture and sound recordings with alphanumeric data in an annotated slide projection.

A reasonable basis for the integration of alphanumeric data and media data is provided by the concept of objects. Different data (alphanumeric as well as bitmap types, for instance) that form a logical unit are composed to form a more complex unit, referred to as composite or complex object. An important feature of objects is the encapsulation of data in the sense that an object hides the implementation of an encapsulated data structure by providing a set of operations on these data structures, representing the only way of accessing and manipulating the data structures. As mentioned above, image or sound data, for instance, that occur as bitmaps, require registration data in order to enable a proper interpretation of the bitmaps. Hence, it is quite natural to combine these registration data and bitmaps into complex objects, generally referred to as media data objects or media objects, for short. For the combination of different media objects the terms multimedia object or mixed media object have been coined. Efforts are being made to develop multimedia database management systems [WK87, MLW88] that allow the users to connect media objects to alphanumeric data, e.g. a photo of a person and his/her name, birth date, address, etc. Thus, a media object is a complex object by nature as it is necessarily composed of subordinate data objects of simple or complex types. Objects that contain only raw data of a single medium are referred to as media objects. Such media objects, containing either raw data of the same medium or of different media, can be combined to form a new, more complex object. This resulting object generally is called a multimedia object.

Regarding the environment in which a multimedia object system may be needed, it is very likely that it will be a heterogeneous system. This development can be observed in almost every computerized application domain. That implies the coexistence of many different media object models. For the time being there is already a considerable variety of different data models for any type of media object available, i.e. there are lots of image, text and graphics data models in the market. One may question, then, why is another model needed. The answer to that is that these models are too application specific and are not generalizable. The operations on media objects are extremely application specific and should therefore form a part of the application itself and not of a media object management system. We need a media object model on a conceptual level that takes into account the uniformness of media objects from the structural point of view. This model must be flexible and extensible to provide a reasonable basis for integrating media objects into multimedia objects.

Hence, in the rest of this paper we will discuss a generalized model for media objects that supports the combination of alphanumeric data and media data in the two ways as described above and that fulfills the requirement of extensibility and flexibility. We start with an overall description of our media object model DEMOM (DEscription based Media Object data Model) which is supposed to be applicable to the different types of media data as listed above, followed then by a brief description of the media object data types that are specific for the different media. A focal point of interest is content-oriented search for media data which we consider to be crucial .

2.0 THE DEMOM MEDIA OBJECT MODEL

Two ways are generally in use for the management of complex objects: either one can extend an existing data model towards complex objects as has been done with the relational model in POSTGRES [SR86], for instance, or one takes an object oriented approach like [WK87]. In the first case, the requ'rement of providing support for conventional and non-standard applications is basically satisfied. However, there are further operations required that are not provided by the relational model and must be developed for such a system. The second approach is based on an object oriented DBMS that is able to store objects of arbitrary length. In currently existing systems (c.f. [MSOP86]) the set of operations defined for manipulating complex objects is very general in nature and does not take into account of specific requirements of certain object types. Thus it is still up to the "user" of the object oriented DBMS (here the application programmer) to write the functions that are needed for managing the objects of a specific application under consideration.

2.1 Basic Model

As outlined in the introductory part, one needs a data model for complex objects that is not bound to a certain type of medium but is general enough to be uniformly applicable to different types of media objects. As a consequence, we have concentrated on a model in which a media object consists of three parts:

- registration data,
- raw data,
- content-description data.

The registration data contain the object identification, a set of common registration data like ownership and access rights and a set of media specific information like a colormap, height, width, and pixel depth for an image, or sampling rate and encoding type for a sound, for instance. Although the internal structure of media data heavily depends on the media type, registration data as such are necessary for all media to enable the correct interpretation of the raw data. The raw data section is a bitmap whose internal structure is disregarded.

The content-description data section, or description for short, contains a natural language description of the object represented by the raw data. This description aims at supporting content search of the media data.

In order to preserve the consistency of related registration data, raw data, and description these different data sections are encapsulated. That is a media object represents an abstract data type that is only accessible through media object type specific operations (figure 2-1).

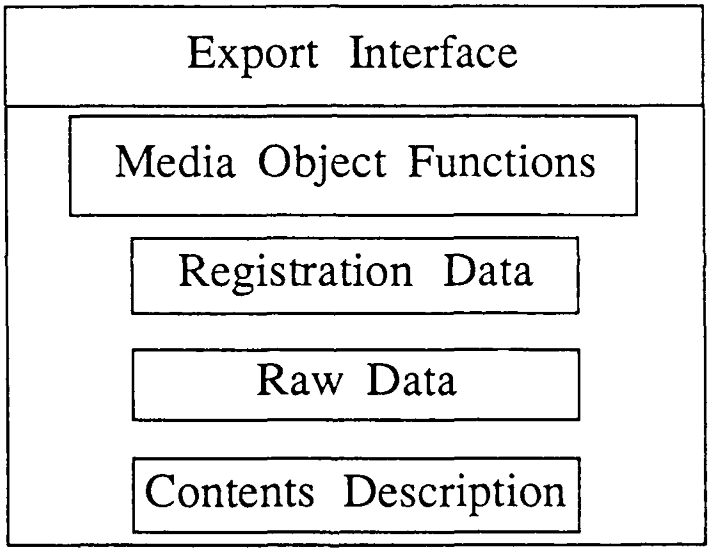

Figure 2-1: Media Object Model

2.2 Object Oriented Representation of Media Objects

In terms of an object oriented approach DEMOM can be characterized as follows:

> A media object mo_i is an instance of a class MO that consists of a name, common registration data, raw data, and a description. The class MO provides the aforementioned methods for manipulating the components of class members.

Media type specific classes can again contain subclasses that represent specific formats of the corresponding media type. For each media type there exists a wide variety of different formats. Almost every WYSIWYG (What You See Is What You Get) providing text processing tool has its own internal representation format. The same holds for graphics tools and for images that depend more on the underlying hardware. All such formats are considered as subclasses of the media type class they belong to. The subclasses thus inherit the structure from their superclasses and, in addition, have their own part of registration data that are necessary for the interpretation of the raw data.

Figure 2-2 illustrates a sample MO class hierarchy. The acronyms IMG, SND, TXT, and GRP stand for the media object subclasses of images, sound, text, and graphics, respectively. IMG, SND, TXT or GRP are subclasses of MO, i.e. their relationship to MO is characterized as an 'IS_A'. Instances of these subclasses inherit the structure of MO (type inheritance) and have in addition a component that contains media subtype specific registration data. For the manipulation of these additional data a corresponding set of methods is provided.

Furthermore, figure 2-2 shows the data structures encapsulated on each level of the hierarchy. MOID is a media object's systemwide unique identification. MO_type specifies the subclass to which the media object belongs. MO_Name contains a symbolic name a user may assign to a media object. Ownership or access rights and the like are regarded as typical registration data (Common_RegData), common to all types of objects. RawData contains the real raw media data part. A natural language based description of a media object's content is maintained in DescrData.

The IMG subclass is further detailed into the classes PIX, ALV and URL. PIX stands for the SUN/Pixrect format [SUN86], ALV is a raster image format developed at Brown University, and URL stands for Utah Run Length Encoded, an image format developed at the University of Utah [PBT86].

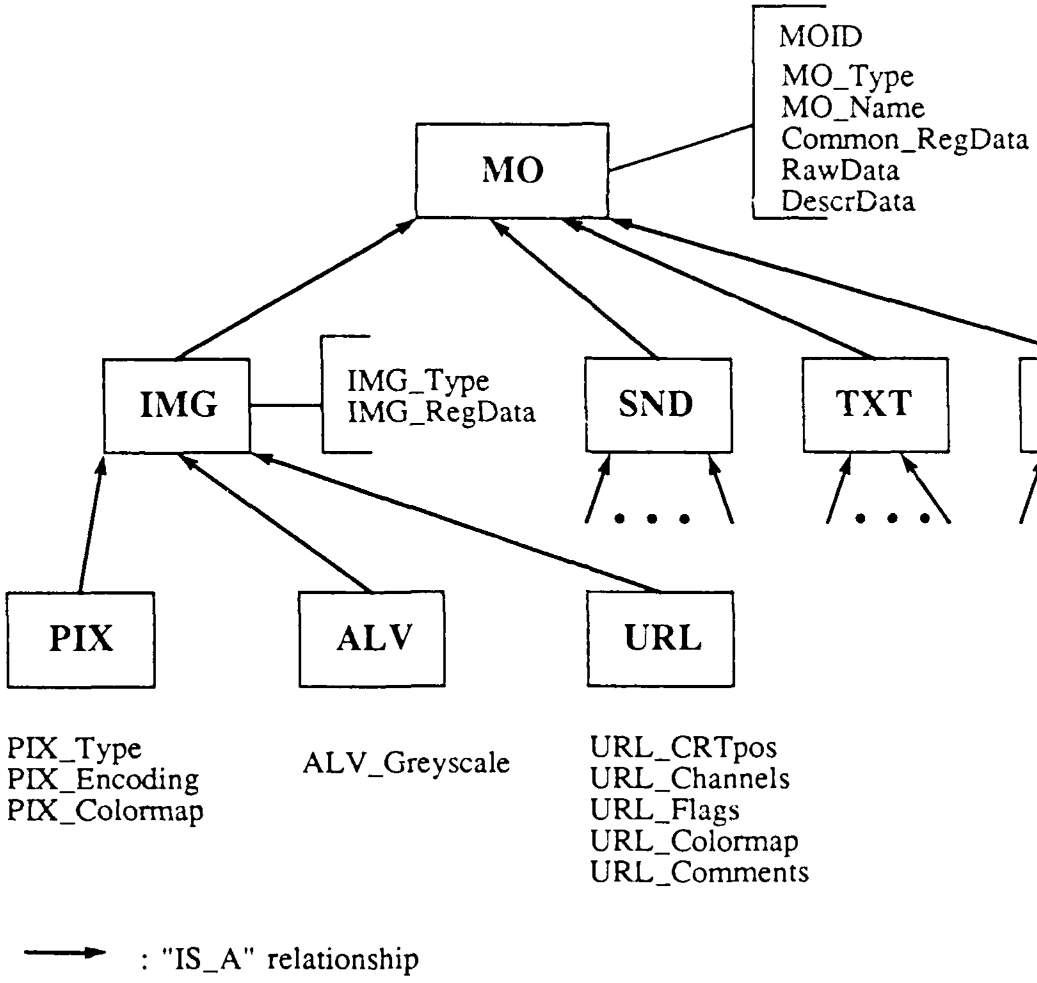

Figure 2-2: Sample Media Object Class Hierarchy

In addition to the common registration data a media object type, i.e. a subclass of MO, has its specific registration data. Typical examples for IMG specific registration data are the height, width and pixel depth of an image and for SND objects are the sampling rate. These registration data are included in the IMG_RegData or SND_RegData components, respectively.

For the subclasses of IMG some subclass specific registration data are listed. They provide colormap information, for instance, and other representation specific information. Thus, the registration data section, shown in figure 2-1, is made up of common registration data, media type specific registration data, and of format specific registration data.

The hierarchy described here is not restricted to the form depicted. Instead, new media types can be added. Potential candidates, for instance, are signals and videos [Lo88]. Due to the high level of abstraction of DEMOM such kinds of media objects can be integrated readily.

2.3 Methods on Media Objects and Subclasses

We assume that a media object is identified by a systemwide unique object identifier MOID of type moid. All operations have this identifier as their first argument.

class MO (subclass of OBJECT)

void MO_remove (MOID)

The media object denoted by MOID is removed from the set of media objects.

void method MO_add_descr (MOID, *char)

This method adds a description, pointed to by *char, to the media object, identified by MOID.

void method MO_replace_description (MOID, *char)

This method replaces the description of media object MOID through the description, pointed to by *char.

*char method MO_get_description(MOID)

For the media object denoted by MOID the related natural language description is returned.

The following is a group of methods that operate on common registration data. As long as we have not precisely defined what the common registration data are we use non-terminal symbols for the definition of the method group.

<commonregdata> method MO_get_<commonregdata> (MOID)

This set of methods returns common registration data of a media object MOID. As potential registration data that are common to all media objects we envisage access rights, ownership, date of creation, date of last modification and the like.

As a counterpart, we envisage the following group.

void method MO_set_<commonregdata> (MOID, <commonregdata>)

This set of methods enables the modification of common registration data.

Finally, we have a set of boolean methods, each of which is related to a subtype of MO. The method returns the value 'TRUE' if the object identified by MOID is of the type that is checked.

```
boolean method MO_IMG (MOID)
boolean method MO_SND(MOID)
   ...
```

In order to enable operations on media objects of different types we support methods on the class MO, or more precisely methods on sets that belong to the class MO (c.f. GemStone and OPAL [MSOP86]). In particular, we think of content based retrieval. The example "Give me a list of all media objects that are related to Beethoven" illustrates this kind of application. The multimedia system can contain a picture of Beethoven, a book or letters of Beethoven, and a set of sound recordings of Beethoven's music, for instance.

```
class MO_SET (subclass of SET)
```

sel^e^ct (mo_setid1, [x: MO_contents(x, Description)])

> By means of the Description argument that contains the natural language description of the objects that the caller is looking for, a (maybe empty) set of media objects is identified and the list of identifiers is returned.

select (mo_setid2, [x: MO_by_<commonregdata>(x, <commonregdata>)]

> This set of methods has as parameter a pointer to the input value for the corresponding common registration data type. Thus a set of media objects can be retrieved by means of their registration data. The (possibly empty) list of identifiers is returned by these methods.

The subclasses of MO (i.e. IMG, SND, TXT and GRP) inherit these methods and add on their own, MO type specific ones. As an example we regard image specific methods on image specific registration data.

Raster images are characterized through their height, width and pixel depth, i.e. the number of bits used for the color or greyscale definition of a pixel. Consequently, these information constitute an image's registration data. That in turn means that we have to provide methods on these data if we intend to use them for retrieval, for instance. As these information are tightly coupled with the raw data part of an image, we cannot modify the registration data without modifying the raw data and vice versa. As a consequence, we provide read-only operations and leave the consistent update of raw data and registration data to the application, i.e. update-in-place is not supported, instead a new version of an image has to be created.

class IMG (subclass of MO)

```
int method IMG_get_height (MOID),
int method IMG_get_width (MOID),
int method IMG_get_depth (MOID)
   ...
```

These methods return the height, width or pixel depth, respectively, of the image denoted by MOID.

In a similar way we can define methods on media object type specific registration data of SND, TXT or GRP objects.

Specific formats of these media object types are again subclasses that inherit the methods defined for the corresponding type and add to their format specific methods. The specific format materializes again as an extension of the registration data part that is needed for the proper interpretation of the specific raw data format. Typical information on this level are encoding information, i.e. text formatting information or data compression information. Referring to the IMG subclasses, depicted in figure 2-2, we see that typical registration data on this level are colormap or greyscale information and encoding techniques used. For these data it is difficult to determine on beforehand whether updates can be allowed or not. The encoding technique used, for instance, is again tightly coupled with the raw data and hence updating it without modifying the raw image would hurt the object's consistency. Colormap information, on the other hand, could be modified without modifying the corresponding raw image. In this case it depends on the application whether update possibilities are desired or not. As a consequence the methods defined on this level can only have the character of an example, not more. Therefore, we restrict ourselves to the presentation of methods on images in pixrect format.

class PIX (subclass of IMG)

ENCODING method PIX_get_encoding (MOID)

This method returns the encoding type used, i.e. RT_OLD, RT_STANDARD or RT_BYTE_ENCODED.

int method PIX_get_cmap_len (MOID)

This method returns the colormap length of image MOID.

int method PIX_get_cmap_entrysize (MOID)

This method returns the size of an entry in MOID's colormap in bytes. A common size for RGB encoded images is 3 bytes, for instance. That provides for 256 different colors.

cmap method PIX_get_cmap (MOID)

This method returns the entire colormap of a pixrect image.

void method PIX_put_cmap (MOID, cmap)

As a counterpart to the preceding method, this one substitutes the current colormap of MOID by the one given as parameter cmap.

This list is not complete but aims at giving an idea how the methods on the lower level look like. As in object oriented systems in general, this list can easily be extended or adapted to the specific needs of a particular application. In a similar way the methods for other media types can be defined.

3.0 CONTENT SEARCH

Storing media data in a computer is not a problem; how to query the content of this data is. For example, if a witness wants to search the digitized criminal mug-shots to identify a suspect that has mean-looking face with a banana nose and beady eyes, the digitized images alone will not help much because extracting those features of an image is complex. A major difficulty in handling multimedia data is the richness of its semantic content. A number 100 associated with the attribute of 'loan balance' can mean very few things. An image of a person, on the other hand, implicitly contains a great deal of information. For images in natural settings, the implications are still more complex, and video data still more.

As already indicated by the description component of DEMOM we follow the approach of contents based search by means of verbal descriptions that form a part of a media object [LM89]. That does not mean, however, that we limit ourselves to this approach exclusively.

3.1 Rationale for Limited Natural Language Descriptions

A well-known approach to content description is the keyword approach as done in library information retrieval. But keyword search techniques have been demonstrably imprecise, except for simple applications, and users have often had great difficulty in focusing the search to documents of interest. The problem of keyword search is that keywords are discrete and no association between keywords are specified.

Graphics objects, in general, have an internal structure (e.g. boxes, circles etc.) so that pattern matching algorithms become applicable. The same is true for relatively simple structured image and sound objects. In such cases it might be appropriate to use pattern matching algorithms.

The problem with image, sound, and graphics objects is that if the objects under consideration have very complex structures and are rich in semantics tools for contents analysis become very complex and extremely time consuming and are thus not suitable for retrieval in a multimedia database environment.

The attachement of keywords also to non-textual data objects is a first step that enables the application of text retrieval mechanisms also to non-textual data. As pointed out above, keywords lack more complex linking mechanisms to adequately capture the contents of objects like aerial photos, for example. Hence, the use of natural language descriptions seems to be a more viable solution. Full understanding of natural languages is not yet achievable, but caption understanding needs only a subset of its techniques. The studies in the areas of natural-language understanding, expert and rule-based systems, and knowledge representation and organization are utilized to help reduce the content search problem in multimedia data to the simpler caption-analysis problem. Captions are a natural but special, stylized way of writing descriptions with a subset of natural language.

Hence, our approach aims at applications where the aforementioned strategies fail, i.e. where media objects contain a high degree of semantics that cannot or at least is not yet covered by the current more "syntax-oriented" approaches. It is to the media DBMS architecture to provide for the coexistence of different retrieval strategies. As a major feature of object oriented systems is the extensibility of the set of methods that goes with an object, we can easily imagine to have "syntax-oriented" methods and semantics based methods together in the same object.

3.2 Scope Limitations of Natural Language Descriptions

Natural language descriptions have the advantage that everyone is familiar with a natural language and therefore one can expect low resistance to the acceptance issue. But that does not automatically solve the problems of description understanding and matching of descriptions and queries. To be successful, focusing on specific application domains is necessary. Narrowing down on a particular application generally is accompanied by restricting the universe of discourse. Further, making assumptions about the user or limiting the capabilities of the interface the user is provided, enables us to reduce the variety of syntactical structures that the natural language processing component must be able to handle.

3.2.1 General Syntax Restrictions

As descriptions are to provide us with facts about the raw data content, one does not need the full capability of a natural language. Hence, we made several restrictions with respect to the grammar of captions that our system accepts.

The first restriction is to limit the natural language to declarative statements. Remember, the reason for natural language processing in this context is the advantage of content description compared to object recognition or keywords. That does not automatically imply the use of all types of grammatical structures. Descriptions in general are of declarative nature.

Further, as style is not a matter of importance, we limit the declarative statements to be active voice and to be entirely composed of certain grammatical structures. Such way of restriction is not believed to cause handicaps or the loss of power in describing the contents of the data but reduces the problem to solve.

Another restriction is related to pronouns. As the system is supposed to be used by multiple users we can limit the use of pronouns and verbs to 3rd person singular and plural.

In addition to these basic restrictions we also defined a grammar that limits the flexibility that is normally available in a natural language. Occurrences of prepositional phrases and participial phrases, for instance, are strictly regulated. A detailed discussion of our grammar is beyond the scope of this paper. For a complete description we refer to [Du90].

3.2.2 Limiting the Universe of Discourse

As a database generally is restricted to a specific application, the vocabulary and the interpretations of them, as well as the interpretation of the descriptions and their semantics, are naturally constrained to a narrow domain of discourse. This means the system does not have to be able to understand everything that can be written with a natural language. The model and the knowledge needed for understanding are therefore much smaller and become manageable. To handle this problem our approach bases on an extended feature based dictionary (c.f. [GM89]), in which the users define the domain of each application, thus restricting their vocabulary and meanings, the semantics, the knowledge and the model for the system to apply.

3.3 Coping with Locations and Time

Captions make extensive use of place and time descriptions. Thus, natural-language understanding of captions requires detailed hierarchies of place names and time interval names. The hierarchy will necessarily be tangled, since there can be alternative generalizations of a term. Efficient access to place and time interval names is crucial since there are many ways to site the same image in time and space, and it will be infrequent that the "standard description" will be identical to what the querier of a system wants to call it.

Hierarchical relations between location names are stored as parts of the location name entries in the dictionary. At description processing time these relations are resolved by means of inference rules. Thus a query like "show me all aerial photos of Californian cities" results in a displaying photos of San Francisco, Los Angeles, San Diego etc. as their dictionary entries say that they are a 'PART_OF' California.

Time expressions are treated in the same way.

3.4 Description Predicates and Their Representation

Since natural-language captions require time and an arsenal of techniques to analyze, we parse captions during entry of their associated media objects into the database, and store the parse results in the database for access to the media objects. Since most parsing methods create a predicate calculus expression representing the meaning of some natural language (c.f. [Wi84], [GM89]), which almost always is the logical conjunction of a large number of terms, we represent the meanings of captions as lists of literals in Prolog notation. Inclusion of a parser in our system also means we can accept queries in natural language to the database. Then their descriptions can be matched to the descriptions of all media objects in the database to find all matches to the query.

The imprecision and ambiguity of the natural language descriptions is reduced considerably by transforming them into a set of predicates. These predicates state facts about the real-world objects involved in the media object. Real-world objects and activities are referred to through their name, or identifier. The predicate states their properties - as indicated by the media object - and their relationships. In many cases, the name of the object may not be known, so that artificial identifiers have to be created. For instance, an image showing a car can be described by the predicates "car (x), manufacturer (x, Horch), year_built (x, 1922)", etc. The use of the name connects different predicates that state properties of the same object.

The parser uses a dictionary of all the words it can recognize, and this dictionary also shows the predicates to use when a word appears in the description. Hence, the set of all predicates that can be used in the descriptions, is defined in the dictionary.

3.5 Matching

The result of parsing is one set of predicates per media object, interconnected by object identifiers. A query is also entered in natural language and then parsed. In contrast to the description, the arguments of the query predicates can be variables. A media object is selected for the result of the query, if and only if there exists a binding of those variables to object identifiers such that the description predicates of the media object logically imply all the query predicates.

The match of user query to database media object need not be exact. A set of rules, some of which must be domain dependent, specifies situations in which sets of literals that look different are really the same. Reasoning about type hierarchies is a simple but important example; for instance, a query that specifies a road should match a caption that specifies a freeway. However, the inverse may not be valid, i.e. freeway is a special kind of road and the distinction may be necessary. Another example is inference of containment of one time interval in another; for instance, a query that mentions a date between May 15th and June 30th should match a caption that specifies June 1st. Another important example is reasoning about physical relationships; for instance, a query that mentions a forest west of a road should match a caption that specifies a road east of a forest.

The matching catches different natural language phrases with the same meaning, but it does not catch semantic relationships among predicates. If the description for an image is "a car with a red body", the predicates generated will be something like "car (x), component (x, y), body (y), color (y, red)". A query that asks for "a red car" is translated into something like "car (x), color (x, red)", and there would be no match. The system does not know that the color of a car's body is just the same as the color of the car.

To overcome this problem, rules can be introduced that express the semantic relationships among the predicates. In our example, the rule could be:

if car (A), component (A, B), body (B), color (B, C) then color (A, C).

Using this rule, color (x, red) can be inferred in the example, and thus the query would match the description. This is similar to the use of S-rules in the START system [Ka88].

A key unsolved problem, only really serious with multimedia data, is which literals to try to generalize to get a match, and how far to generalize. Domain-dependent knowledge can help, but we believe there are undiscovered general principals.

4.0 CONCLUSION

The handling of multimedia data imposes new requirements on database management systems, especially when regarding the integration support of conventional and multimedia data. In this paper we present an approach that aims at easily integrating conventional alphanumeric and multimedia data by providing the object-oriented DEMOM media object model.

A media object is designed as encapsulating raw data, registration data and, in contrast to other projects in the area of multimedia DBMS research, contents-description data. These data can only be accessed through object specific operations that also form a part of the entire media object.

The raw data contain the byte stream whose potential internal structure is disregarded. Thus, for the reasonable interpretation of the raw data some registration information is necessary. The additional contents-description data are introduced for supporting contents search. Internally, these descriptions are translated into description predicates by means of a predicate calculus.

Due to the general nature of our object model it is applicable to the different kinds of media without modification. Hence we designed our multimedia DBMS as being composed of a set of media DBMS which have a uniform gross architecture (c.f. [Lo88]). The multimedia DBMS architecture sketched there looks very similar to that of the MUSE multidatabase system [Ho88] that is implemented as a decentralized system. The correspondence in the architecture and the flexibility of the MUSE system makes it an ideal candidate for the integration of the media DBMS. As described in [Ho90] MUSE's transaction concept also provides the flexibility and extensibility that is required for multimedia object management. Consequently, on this basis we can also envisage the multimedia DBMS as being distributed.

At the time being we have a prototype implementation for image objects in an image DBMS [Th88]. The prototype supports images in SUN pixrect format and in ALV format. The system is running on SUN under OS 4.0.1. It is written in C and uses the Ingres relational database system for managing the registration data of image objects.

A major research issue that will be tackled in the future is the management of description predicates. Logically the predicates are stored and indexed in the database to facilitate query processing. The data structure that is best for this purpose is not known. Normal indexing method used in conventional database systems where each predicate is followed by a list of data instances containing that predicate are found to be inadequate, as some of the predicate terms may not be very selective--i.e. some predicates may be associated with a very large number of media data instances. Trade-offs involving detailed data structures and the processing strategies must be analyzed carefully to draw any concrete conclusions.

ACKNOWLEDGMENTS

We wish to thank Klaus Meyer-Wegener for his contributions to an earlier version of the multimedia DBMS concepts and Peter Lockemann for discussions and comments on an earlier version of this paper. Thanks also to Cathy Thomas and Greg Sawyer for their implementation work on media DBMS components. We are also endepted to John Dulle who defined the grammatical scope of our caption approach.

REFERENCES

[Ka88] Katz, B. "Using English for Indexing and Retrieval", Proc. RIAO88, Cambridge (MA), March 1988

[Du90] Dulle, J. , "The Scope of Descriptive Captions for Use in a Multimedia Database System", M.S. Th., Naval Postgraduate School, Department of Computer Science, Monterey, CA, in preparation

[GM89] Gazdar, G. and Mellish, C., "Natural Language Processing in PROLOG", Addison-Wesley, 1989

[Ho88] Holtkamp, B., "Preserving Autonomy in a Heterogeneous Multidatabase Environment", Proc. COMPSAC'88, Chicago (Il.), Oct. 1988

[Ho90] Holtkamp, B., "Flexible Cooperation in a Non-Standard Application Environment", Report no. NPS52-90-018,Naval Postgraduate School, Department of Computer Science, Monterey, CA

[LM89] Lum, V.Y., and Meyer-Wegener, K., "A Multimedia Database Management System Supporting Content Search in Media Data,"Report no. NPS52-89-020, Naval Postgraduate School, Department of Computer Science, Monterey, CA. March 1989.

[Lo88] Lockemann, P.C. "Multimedia Databases: A Paradigm and Architecture", Report no. NPS-52-88-047, Naval Postgraduate School, Department of Computer Science, Monterey CA, Sept. 1988

[MLW88] Meyer-Wegener, K., Lum, V.Y., and Wu, C.T., "Managing Multimedia Data,"Report no. NPS52-88-010, Naval Postgraduate School, Department of Computer Science, Monterey, CA, March 1988.

[MSOP86] Maier, D., Stein, J., Otis, A. and Purdy, A., "Development of an Object-Oriented DBMS", Proc. OOPSLA'86

[PBT86] Peterson, J.W., Bogart, R.G. and Thomas, S.W., "The Utah Raster Toolkit", Proc. 3rd Usenix Workshop on Graphics, Monterey (CA), Nov. 1986

[SR86] Stonebraker, M. and Rowe L., "The Design of POSTGRES", Proc. SIGMOD Conference, Washington D.C., May 1986

[SUN86] Sun Microsystems, *Pixrect Reference Manual*, Part No. 800-1254-03, Revision A, Mountain View, CA, Febr. 1986.

[Sh88] Sheth, A. "Managing and Integrating Structured and Unstructured Data: Problems of Representation, Features, and Abstraction" Proc. 4th Int. Conf. on Data Engineering, Los Angeles, Feb. 1988

[Th88] Thomas, C.A., "A Program Interface Prototype for a Multimedia Database Incorporating Images", M.S. Th., Naval Postgraduate School, Department of Computer Science, Monterey, CA, December 1988.

[WK87] Woelk, D. and Kim, W. "Multimedia Management in an Object-Oriented Database System", Proc. 13th Int. Conf on VLDB, Brighton (England), Sept. 1987

[Wi84] Winston, P.H., "Artificial Intelligence", 2nd edition, Addison-Wesley, 1984

[Yo88] Yost, R.A. "Can image data be integrated with structured data?" Proc. 4th Int. Conf. on Data Engineering, Los Angeles, Feb. 1988

Distribution List

SPAWAR-3242 Attn: Phil Andrews Washington, DC 20363-5100	1
Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
Dudley Knox Library, Code 0142 Naval Postgraduate School Monterey, CA 93943	2
Center for Naval Analyses 4401 Ford Ave. Alexandria, VA 22302-0268	1
Director of Research Administration Code 012 Naval Postgraduate School Monterey, CA 93943	1
John Maynard Code 402 Command and Control Departments Naval Ocean Systems Center San Diego, CA 92152	1
Dr. Sherman Gee ONT-221 Chief of Naval Research 800 N. Quincy Street Arlington, VA 22217-5000	1
Leah Wong Code 443 Command and Control Departments Naval Ocean Systems Center San Diego, CA 92152	1

Bernhard Holtkamp 5
Code 52Hk
Naval Postgraduate School
Monterey, CA 93943

Vincent Y. Lum 50
Code 52Lu
Naval Postgraduate School
Monterey, CA 93943

Neil C. Rowe 1
Code 52Rp
Naval Postgraduate School
Monterey, CA 93943

Klaus Meyer-Wegener 1
University of Kaiserslautern
Computer Science Department
P.O. Box 30 49
D-6750 Kaiserslautern
West Germany